HEWLETT PACKARD

INSTALLATION MANUAL

12907A

FAST FORTRAN PROCESSOR HARDWARE ACCESSORY KIT

(FOR 2100 COMPUTERS)

Note

This manual should be retained with the applicable computer system documentation.

12907-90001

HEWLETT-PACKARD COMPANY
11000 WOLFE ROAD, CUPERTINO, CALIFORNIA, U.S.A.

PRINTED: JAN 1975

CONTENTS

ILLUSTRATIONS

TABLES

1. INTRODUCTION.

2. This manual provides field installation instructions for the HP 12907A Fast Fortran Processor Hardware Accessory Kit, an accessory for Hewlett-Packard 2100 Computers. Additional information regarding Hewlett-Packard 2100 Computers is provided in the associated computer documentation as follows: for theory of operation and maintenance information, refer to the *Installation and Maintenance Manual;* for schematic and component location diagrams, refer to the *Diagrams Manual;* for replacement parts information, refer to the *Illustrated Parts Breakdown Manual;* and for a description of programming requirements and sample assembly language programs, refer to the *Reference Manual.*

3. DESCRIPTION.

4. The HP 12907A Fast Fortran Processor Hardware Accessory Kit provides the computer with thirteen subroutines implemented in firmware, i.e., a read-only-memory (ROM) microprogram. These include the five Fast Fortran subroutines, i.e., .GOTO, . .MAP, .ENTR, .ENTP, and .SETP and the eight Extended Precision subroutines, i.e., DBLE, SNGL, .XMPY, .XDIV, .DFER, .XFER, .XADD, and .XSUB. The firmware version of these subroutines is used to replace the existing software version and thus increase execution speed.

5. The microprogram for the Fast Fortran and Extended Precision subroutines are contained in twelve ROM integrated circuits which must be properly installed in modules 2 and 3 on Timing and Control Card A1. The socket locations for module 2 are U12, U22, U32, U42, U52, and U62, while U11, U21, U31, U41, U51, and U61 are the socket locations for module 3. Mapping to the correct ROM starting address for the execution of these subroutines is accomplished by the proper configuration of jumpers W4 and W5 in the ROM mapper circuits on ROM Control Card A2. The configuration of jumpers W1, W2, W3, and W6 in the Non-Existent ROM (NER) FF circuits is also changed to include ROM modules 0, 1, 2, and 3. The HP 12901A Floating-Point Hardware Accessory Kit must be installed in sockets U25, U26, U27, U35, U37, and U65 on ROM Control Card A2 (module 1) when this accessory is used.

Note: The HP 12908B Writeable Control Store (WCS) Interface Kit cannot be used once the Fast Fortran Processor has been installed, since all module positions are in use.

6. The HP 12907A Fast Fortran Processor Hardware Accessory Kit consists of the following:

a. Twelve ROM integrated circuits packages, part numbers 1816-0368 through 1816-0373 and 1816-0490 through 1816-0495.

b. Twelve 16-pin integrated circuit sockets, part number 1200-0482.

c. Twelve socket insulators, part number 0340-0788.

d. One 50-pin interconnecting cable, part number 02100-60028.

e. One product identifier, part number 12907-80001.

f. One *Instruction Manual,* part number 12907-90001.

g. One subroutine library update binary tape, part number 12907-16001.

h. One $SETP system update binary tape, part number 12907-16002.

i. One diagnostic binary tape, part number 12907-16003.

7. When this accessory kit is to be installed in an existing computer system, Microinstruction Decoder Card A4 must be checked to verify that it is of the proper configuration, i.e., part number 02100-60112. If the part number etched on the A4 card presently installed in the computer system is 02100-60022, it must be replaced with option 001, i.e., part number 02100-60112.

8. UNPACKING AND INSPECTION.

9. If the shipping carton is damaged upon receipt, request that the carrier's agent be present when the contents of the kit are unpacked. Inspect the contents for damage (cracked or broken parts, etc.). If the contents are damaged or fail to meet specifications, notify the carrier and the nearest HP Sales and Service Office immediately. Sales and Service Offices are listed at the back of this manual. Retain the shipping container and the packing material for the carrier's inspection. The HP Sales and Service Office will arrange for the repair or replacement of the damaged components without waiting for any claims against the carrier to be settled.

10. IDENTIFICATION.

11. Hewlett-Packard uses five digits and a letter (00000A) for standard kit designation. If the designation of this kit does not agree with that on the title page of this

manual, there are differences between the kit and the kit described in this manual. These differences are described in change sheets and manual supplements available at the nearest HP Sales and Service Office. These offices are listed at the back of this manual.

12. Printed-circuit card revisions are identified by a letter, a series code, and a division code stamped on the card, e.g., A-0000-00. The letter code identifies the version of the etched trace pattern on the unloaded card. The series code (middle four digits) refers to the electrical characteristics of the loaded card and the position of the components. The division code (last two digits) identifies the Hewlett-Packard division that manufactured the card. If the series code stamped on the printed-circuit card does not agree with the series code shown in the appropriate schematic diagram in the computer *Diagrams Manual*, there are differences between the card and the card described in the *Diagrams Manual*. These differences are described in manual supplements available at the nearest HP Sales and Service Office.

13. INSTALLATION.

14. Install the HP 12907A Fast Fortran Processor Hardware Accessory Kit as follows:

a. Turn off power at the computer.

b. Remove the top access cover from the computer.

c. Remove all cable connectors (if any) attached to the top of Timing and Control Card A1, ROM Control Card A2, and Microinstruction Decoder 2 Card A4 in slots 1, 2, and 4, respectively. See figure 1, view A.

d. Remove Timing and Control Card A1 from slot 1 and install the twelve integrated circuit sockets and socket insulators (supplied) in the positions allocated to U11, U12, U21, U22, U31, U32, U41, U42, U51, U52, U61, and U62. See figure 1, view B.

e. Insert the twelve ROM integrated circuits into the designated integrated circuit sockets as outlined in table 1. Ensure that the ROM's are oriented with the notched end facing toward the 86-pin edge connector. See figure 1, views B and C.

f. Replace the Timing and Control Card A1 (with Fast Fortran Processor ROM's installed) in slot 1 of the computer and install the product identifier over connector A1J2. See figure 1, views A and C.

g. Remove ROM Control Card A2 from slot 2, and configure the jumpers as outlined in table 2. See figure 1, view D.

h. Ensure that the HP 12901A Floating-Point Hardware Accessory Kit is installed in the module 1 position and that the product identifier is in place over connector A2J2. See figure 1, view A.

Table 1. ROM Integrated Circuit Socket Designations

SOCKET	ROM PART NUMBER
U11	1816-0491
U12	1816-0369
U21	1816-0492
U22	1816-0370
U31	1816-0494
U32	1816-0372
U41	1816-0490
U42	1816-0368
U51	1816-0493
U52	1816-0371
U61	1816-0495
U62	1816-0373

Table 2. ROM Control Card A2 Jumper Configuration

MODULES	JUMPERS INSTALLED					
	W1	W2	W3	W4	W5	W6
0, 1, 2, 3	None	None	None	In	None	None

i. Replace ROM Control Card A2 in slot 2 of the computer.

j. Remove Microinstruction Decoder 2 Card A4 from slot 4, and verify that its part number is 02100-60112. If the part number etched on the card is 02100-60022, the A4 card must be replaced with option 001 (part no. 02100-60112).

k. Once Timing and Control Card A1 (with Fast Fortran Processor ROM's installed), ROM Control Card A2 (with jumpers properly configured), and Microinstruction Decoder 2 Card A4 (with part no. 02100-60112) have been replaced in the computer, replace all cable connectors removed in step c.

l. Connect the interconnecting cable (supplied) between A1J1 and A2J1. See figure 1, views A and C.

15. INSTALLATION CHECKOUT.

16. Turn on power at the computer and perform the computer diagnostic tests outlined in the associated *Installation and Maintenance Manual*. Information for loading the diagnostic tapes and performing the tests is also provided in that manual. Also, reference the *Manual of Diagnostics*. When successfully completed, load and perform the

Floating-Point diagnostic tests as outlined in Diagnostic Program Procedure, part no. 02100-90215, contained in the *Manual of Diagnostics.* When successfully completed, load and perform the Fast Fortran Processor diagnostic tests as outlined in Diagnostic Program Procedure, part no. 12907-90003, contained in the *Manual of Diagnostics.* If the diagnostic programs are completed without an error, the installation was properly executed and the Fast Fortran Processor is operating correctly. The top access cover of the computer can be replaced and the disc operating system can be configured. If the Fast Fortran Processor diagnostic program indicates an error, halt the computer, turn off power, and recheck each of the installation steps. Once rechecked, repeat the diagnostic tests.

17. DISC OPERATING SYSTEM GENERATION.

18. A new disc operating system must be generated to incorporate the Fast Fortran Processor. This is accomplished by loading the Subroutine Library Update Binary Tape, part no. 12907-16001 and the $SETP System Update Binary Tape, part no. 12907-16002 during the Program Input Phase of DOS-III disc generation. Refer to Section X, Generating DOS-III, of the *DOS-III Disc Operating System Manual,* part no. 02100-90136 for DOS-III disc generation procedures. For DOS-M, only the Subroutine Library Update Binary Tape, part no. 12907-16001 must be loaded. Refer to the *Generating DOS-M Manual,* part no. 5951-1375 for DOS-M disc generation procedures.

19. SHIPPING AND STORAGE.

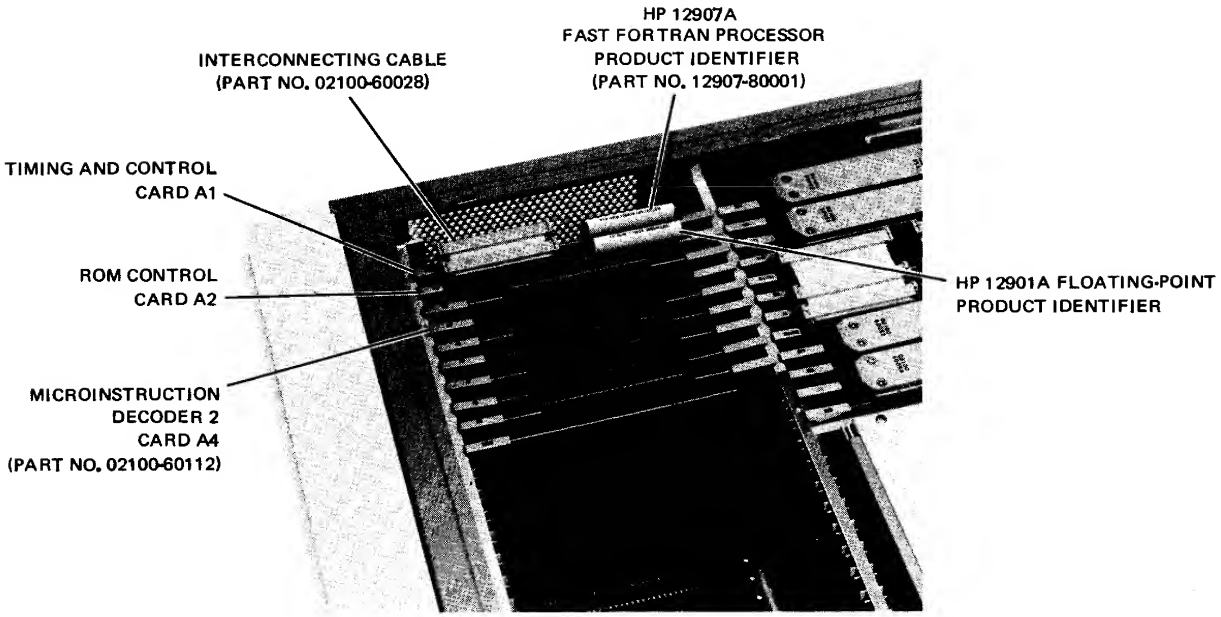
20. If an item from the kit is to be shipped to Hewlett-Packard for service or repair, attach a tag to the item identifying the owner and indicating the service or repair to be accomplished. Include the part number of the kit.

21. Package the item in the original factory packaging material, if available. If the original packaging material is not available, standard factory packaging material can be obtained from a local Hewlett-Packard Sales and Service Office.

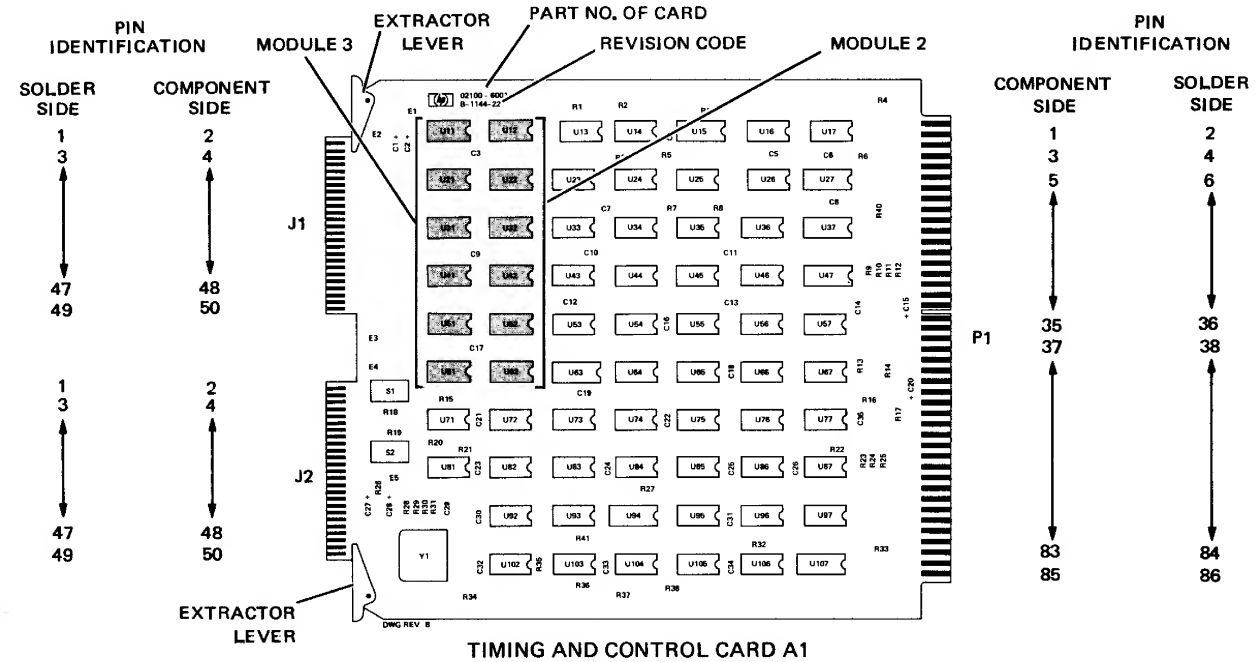
22. If standard factory packaging material is not used, wrap the item in Air Cap TH-240 cushioning (or equivalent) manufactured by Sealed Air Corp., Hawthorne, N.J., and place in a corrugated carton (200 pound test material). Seal the shipping carton securely and mark it "FRAGILE" to ensure careful handling.

Note: In any correspondence, identify the kit by part number. Refer any questions to the nearest Hewlett-Packard Sales and Service Office.

23. If the kit is to be stored before use, package it as previously described to prevent accidental damage.



VIEW A



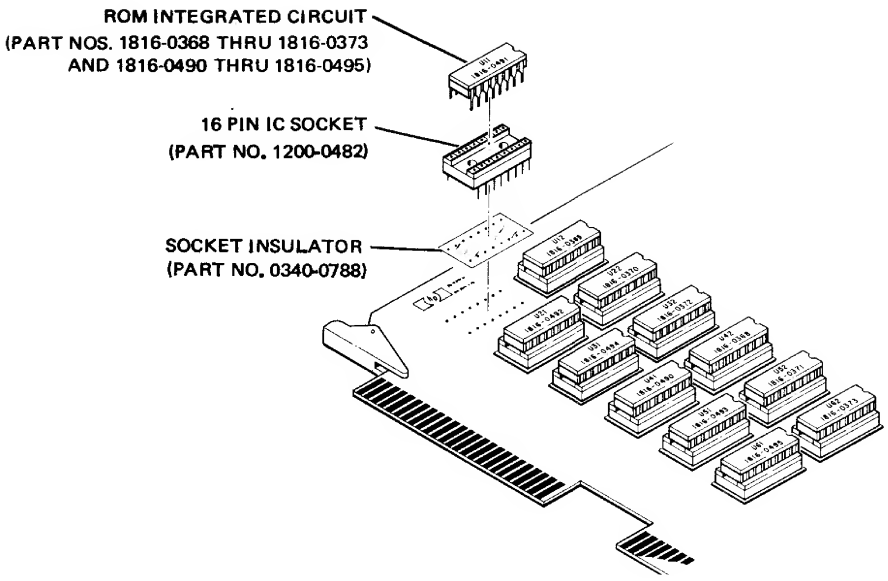
TIMING AND CONTROL CARD A1

VIEW C

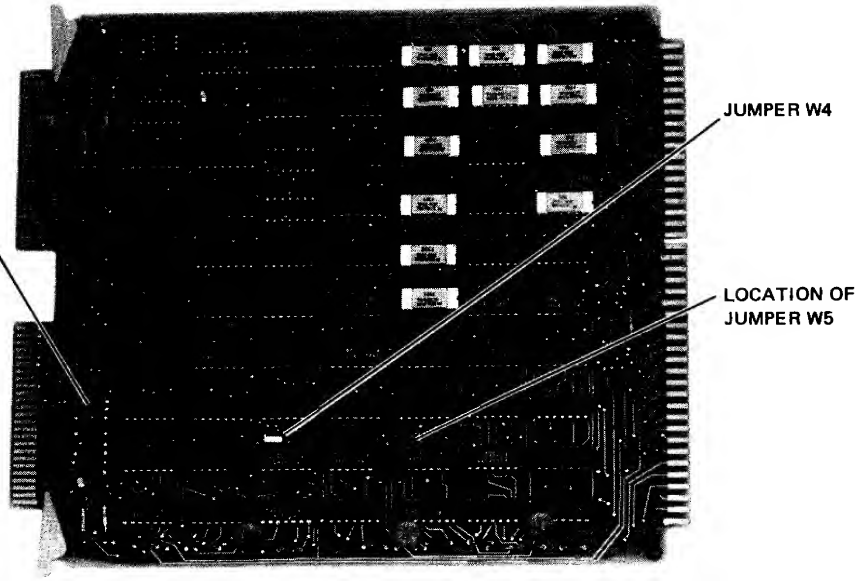
NOTE:

1. INTEGRATED CIRCUITS ARE NUMBERED IN ROW AND COLUMN ORDER.
2. ROM'S MUST BE ORIENTED WITH NOTCHED END FACING TOWARD P1.

7066-1

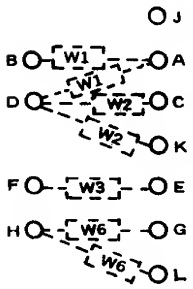


VIEW B



ROM CONTROL CARD A2

VIEW D



DETAIL A

Figure 1. Installation Diagram

HEWLETT PACKARD

2100 series computers

implementing the HP 2100 fast FORTRAN processor

HP Product No. 12907

HP MANUAL PART NO. 12907-90010
MICROFICHE PART NO. 12907-90012

November 1974

List of Effective Pages

Pages	Effective Date
Title	Nov 1974
1	Jan 1974
2	Nov 1974
3 to 19	Jan 1974

Implementing the HP 2100 Fast FORTRAN Processor

The HP 12907 Fast FORTRAN Processor (FFP) consists of 13 microcoded subroutines. These firmware subroutines replace existing software relocatable subroutines, increasing program execution speed from 2 to 20 fold and saving up to 750 words of memory.

The FFP microprogrammed subroutines are fused into programmable read only memory chips and installed in Control Store modules 2 and 3 in the HP 2100 Computer.

FFP can be used at the assembly language level and in the BCS, DOS-M and DOS-III environments.

Related Documentation

FFP can be used in many software environments. Each environment has its special requirements.

- For a detailed description of the hardware product and installation and checkout procedures, see the *Installation Manual* for the *12907A Fast FORTRAN Processor Hardware Accessory Kit (12907-90001).*
- For a description of DOS-III use and generation, see HP 24307B *DOS-III Disc Operating System (24307-90006),*
- For a description of DOS-M use, see *Moving-head Disc Operating System (02116-91779).*
- For the DOS-M generation procedure, see *Generating DOS-M (5951-1375).*
- For reference to all relocatabel subroutines, see *Relocatable Subroutines (02116-91780).*
- For general reference to the Basic Control System (BCS), see *Basic Control System (02116-9017).*
- For Operating procedures for BCS, see *Basic Control System Software Operating Procedures (5951-1391).*

Firmware Implementation

FFP is normally invoked by a user's program via the calling sequences described in the "Firmware Subroutines" section. The calling sequences execute a JSB to the required subroutine. In order to transfer control to the firmware, FFP library subroutines are provided to intercept the calls which would normally go to software relocatable subroutines.

TRANSFER TO FIRMWARE

The memory location of the JSB is changed to a RAM *aaa* or 105*aaa*, where address *aaa* corresponds to an entry point into the FFP firmware. Once the call has been intercepted by the FFP software routine, the RAM *aaa* is left in memory and executed instead of the JSB. Consequently the software is only called once for each FFP subroutine call. Here is the correspondence between the coded software and the result in memory after the subroutine call is executed.

Coded:	Result After Execution
JSB .GOTO	105221
JSB . .MAP	105222
JSB .ENTR	105223
JSB .ENTP	105224
JSB .SETP	105227
JSB DBLE	105201
JSB SNGL	105202
JSB .XMPY	105203
JSB .DFIV	105204
JSB .DFER	105205
JSB .XADD	105213
JSB .XSUB	105214
JSB .XFER	105220

The user could code up his own "RAM 221B" to call .GOTO firmware.

This is not recommended because his program could only run on an HP 2100 computer with FFP installed.

INSTALLING FFP LIBRARY SUBROUTINES

The library subroutines required to use FFP firmware are the following:

1. DOS FFP Subroutine Library (12907-16001); FFP.*n*; used by DOS-III and DOS-M.
2. DOS FFP $SETP System Subroutine (12907-16002); $SETP; used by DOS-III only.
3. BCS FFP Subroutine Library (12907-16004); FFPB*n*; used in BCS only.

To fully use FFP, both the FORTRAN IV and the Floating Point libraries must be present.

Use of FFP Library Subroutine Tapes

DOS-III. The recommended method is for the first two tapes to be included in DOS-III during the "Program Input Phase" of the DOS generation procedure, described in section 10 of the *DOS-III* manual. The two tapes are loaded after all other library tapes since they replace entry point names already loaded during DOS generation.

```
ERR 08
$SETP
ERR 05
$SETP
*EOT
ERR 05
DBLE
ERR 05
SNGL
ERR 05
.DFER
ERR 05
.XFER
ERR 05
.XMPY
ERR 05
.XDIV
ERR 05
.XADD
ERR 05
.XSUB
ERR 05
.GOTO
ERR 05
. .MAP
ERR 05
.ENTR
ERR 05
.ENTP
*EOT

NO UNDEF EXTS
```

Since previous names in the library are replaced, ERR 05 (duplicate entry point name) and ERR 08 (duplicate program name) are printed at load time. These messages should be noted since they indicate that the non-FFP library subroutine names have been replaced and that system generation may proceed.

While it does not fully utilize FFP and is more awkward for the programmer, it is possible to avoid a new system generation after installation of FFP hardware by adding the DOS FFP Subroutine Library tape to the user relocatable programs at load time. The $SETP tape cannot be used in this way. The method for loading the DOS FFP Subroutine Library tape at program load time is the following:

1. Store the DOS FFP Subroutine Library tape onto disc
2. Whenever the FFP library is required, load it with the user programs:

```
@
:PR,LOADR,2
USERP,FFP,/E
L08
LOADR TERMINATED
@
```

Note: "USERP" and "FFP" are the names of the files in which the user program and the DOS FFP Subroutine Library, respectively, have been stored.

DOS-M. The recommended method is for the first tape to be included in DOS-M during the "Program Input Phase" of DOS generation procedure, described in *Generating DOS-M*. The tape is loaded after all other library tapes since it replaces entry point names already loaded during DOS generation. ERR 05 (duplicate entry point) and ERR 08 (duplicate program name) will appear at load time indicating that the non-FFP library routine names have been replaced.

In the same way as for DOS-III, it is possible to avoid a system generation by adding the FFP library tape to the user relocatable programs at load time. The method for DOS-M is the same as under DOS-III.

BCS. The third tape, BCS FFP Subroutine Library, must be loaded *before* loading any other Relocatable Library. See Section 3, "Relocating Loader", of the *Basic Control System* manual for further information on BCS.

Firmware Subroutines

The 13 microcoded subroutines are contained in modules 2 and 3 of the HP 2100 control store. The following subroutines are microcoded in FFP:

.GOTO	.XADD
.ENTR	.XSUB
.ENTP	.XMPY
.SETP	.XDIV
. .MAP	.DFER
SNGL	.XFER
DBLE	

The subroutine descriptions give the purpose, assembly language calling sequence, approximate timing (executed in microcode only), and size for each of the FFP subroutines. The cycle referred to in the timing description is equal to .98 microseconds.

LIMITATIONS

- The microcode programs can execute no more than 16 levels of indirect addressing.
- Floating Point Hardware must be installed on the computer to execute FFP.
- The A- and B-registers cannot be used as pointers when FFP is called from assembly language programs.

.GOTO

Purpose

Transfers control to the location indicated by a FORTRAN computed GOTO statement
GO TO (K_1, K_2, . . ., K_n)J

Calling Sequence

```
JSB .GOTO
DEF *+N+2
DEF J
DEF K1
DEF K2
.......
DEF KN
```

Return

To K_J; IF $J < 1$ THEN K_1; IF $J > N$ THEN K_N;
B-register = address contained in DEF *+N+2

Size

13_8

Timing

13 cycles + number of indirects traced; non-interruptible

.ENTR AND .ENTP

Purpose

Transfers the true addresses of parameters from a calling sequence into a subroutine; adjusts return address to the true return point.

Calling Sequence

User's Program

```
JSB ENTRY
DEF *+N+1 (Return Address)
DEF P1
DEF P2
  .
  .
  .
DEF PN
(continue)
```

User's Subroutine for .ENTR:

```
BUFF    BSS N (maximum number of parameters)
ENTRY   NOP (entry point to subroutine)
        JSB .ENTR
        DEF BUFF
```

User's Subroutine for .ENTP:

```
BUFF    BSS N
ENTRY   NOP
        JSB $LIBR
        NOP
        JSB .ENTP
        DEF BUFF
```

Return

A-register = Return Address
B-register = BUFF + number of parameters actually passed +1

.ENTR and .ENTP put the true addresses of the parameters in "BUFF" and put the return address in "ENTRY".

If more than N parameters are attempted to be passed, only N will be passed and B points to the subroutine entry point.

If less than N are passed, B-register points to last word in BUFF that had a parameter address stored plus one.

Note: DOS Software .ENTR and .ENTP leave B-register = 0.

Size

32_8

Timing

[12 + 4 (# of parameters) + # of Indirects] cycles; non-interruptible

. .MAP

Purpose

To replace FORTRAN's . .MAP; computes the address of a specified element of a 2 or 3 dimensional array; returns the address in the A-register.

Calling Sequence

For 2 dimensions:

```
    CLA
    LDB   Number of words per variable
    JSB   . .MAP
    DEF   Base Address
    DEF   1st Subscript
    DEF   2nd Subscript
    DEF   length of 1st dimension
```

For 3 dimensions:

```
    CCA
    LDB   Number of words per variable
    JSB   . .MAP
    DEF   Base Address
    DEF   1st Subscript
    DEF   2nd Subscript
    DEF   3rd Subscript
    DEF   length of 1st dimension
    DEF   length of 2nd dimension
```

Return

A-register = The address requested

Size

47_8

Timing

26 cycles for 2 dimensions, non-interruptible; 43 cycles for 3 dimensions, non-interruptible; plus 1 cycle for each indirect traced.

.SETP

Purpose

To set a table of increasing values (for DOS-III, $SETP subroutine is used).

Calling Sequence

```
    LDA Initial value
    LDB Initial Address (Direct Address only)
    JSB .SETP
    DEF COUNT
```

Return

A-register = 0
B-register = last address initialized + 1
Initial Address = Initial value
Initial Address + 1 = Initial value + 1
. . .
. . .
Initial Address + COUNT –1 = Initial value + COUNT – 1
COUNT is not modified.

Size

11_8

Timing

8 + 2 (COUNT) cycles plus one cycle for each indirect traced; non-interruptible

Special Considerations

COUNT may not be negative. If it is, the routine will use that as an indirect pointer to a count.

If COUNT is large enough to cause a wrap around (address becomes negative), *the routine will destroy base page without causing a memory protect violation.*

DBLE

Purpose

To convert a 2 word real number to a 3 word real number.

Calling Sequence

Assembly Language:

```
JSB DBLE
DEF *+3 (actually ignored)
DEF X (three word variable)
DEF Y (two word variable)
```

Note: The two word variable Y *should be in normalized form.*

Return

Operation completed.

Size

15_8

Timing

20 cycles, non-interruptible plus one cycle for each indirect traced.

SNGL

Purpose

To convert a 3 word, double precision variable to a 2 word real variable.

Calling Sequence

Assembly Language:

```
JSB SNGL
DEF *+2 (ignored by routine)
DEF Y (3-word variable)
```

Return

Result in the A- and B-registers.

Size

25_8

Timing

Minimum: 20 cycles; non-interruptible
Maximum: 60 cycles; non-interruptible

Error Conditions

If result condition is:	Then, result returned is:
a. beyond $(1-2^{-23})2^{127}$	a. $(1-2^{-23})2^{127}$
b. within $-(1+2^{-22})2^{-129}$ and 0	b. 0

In each case, overflow is set.

.XADD AND .XSUB

Purpose

Add or subtract two double precision variables; i.e.,

```
X = Y + Z  for .XADD
X = Y - Z  for .XSUB
```

Calling Sequence

```
JSB .XADD or .XSUB
DEF X (result)
DEF Y (1st operand)
DEF Z (2nd operand)
```

Size

106_8

Timing

	.XADD	.XSUB
Minimum:	34 cycles	37 cycles
Maximum:	118 cycles	120 cycles

Interruptible after approximately 25 cycles. If interrupted, subroutine restarts at beginning; maximum time non-interruptible: approximately 91 cycles.

Error Conditions

If results condition is:	Then, result returned is:
a. beyond $(1-2^{-39})2^{127}$	a. $(1-2^{-39})2^{127}$
b. beyond -2^{127}	b. -2^{127}
c. within 0 and 2^{-129}	c. 0
d. within $-(1+2^{-38})2^{-129}$ and 0	d. 0

In each case, overflow is set.

.XMPY

Purpose

To multiply two 3-word, double precision variables, i.e.

```
X = Y * Z
```

Calling Sequence

```
JSB .XMPY
DEF X (result)
DEF Y (operand #1)
DEF Z (operand #2)
```

Size

66_8 words

Timing

Approximately 70 cycles, interruptible at approximately 24 cycles; maximum non-interrupt time is 70 cycles due to worst case normalization.

Special Considerations

If interrupted, routine restarts from the beginning; thus every interrupt adds 24-plus cycles to the execution time.

Error Conditions

If result condition is:	Then, result returned is:
a. beyond $(1-2^{-39})2^{127}$	a. $(1-2^{-39})2^{127}$
b. beyond -2^{127}	b. -2^{127}
c. within 0 and 2^{-129}	c. 0
d. within $-(1+2^{-38})2^{-129}$ and 0	d. 0

In each case, overflow is set.

.XDIV

Purpose

To divide one 3-word double precision variable by another; i.e.,

X = Y/Z

Calling Sequence

```
JSB .XDIV
DEF X (result)
DEF Y (1st operand)
DEF Z (2nd operand)
```

Size

73_8

Timing

Approximately 120 cycles, interruptible at approximately 72 cycles.

Special Considerations

If interrupted, routine restarts from the beginning; thus every interrupt adds 75-plus cycles to the execution time.

Note: In 0.9% of the cases the result will be inaccurate in the least significant bit.

Error Conditions

If result condition is:	Then, result returned is:
a. beyond $(1-2^{-39})2^{127}$	a. $(1-2^{-39})2^{127}$
b. beyond -2^{127}	b. -2^{127}
c. within 0 and 2^{-129}	c. 0
d. within $-(1+2^{-38})2^{-129}$ and 0	d. 0

In each case, overflow is set.

DFER

Purpose

To transfer the contents of 3 word, double precision variable to another variable;

X = Y

Calling Sequence

```
JSB .DFER
DEF X (Destination)
DEF Y (Source)
```

Return

A-register = (address of Y)+4
B-register = (address of X)+4

Size

6 words

Timing

15 cycles, non-interruptible plus one cycle for each indirect traced.

.XFER

Purpose

To transfer the contents of a 3 word variable to another variable; i.e.,

X = Y

Calling Sequence

```
LDA (source address X)
LDB (destination address Y)
JSB .XFER
```

Return

A-register = (address of Y)+4
B-register = (address of X)+4

Size

1 word

Timing

11 cycles, non-interruptible for instruction opcode.

Special Considerations

The software supplied with the FFP package uses the .DFER instruction, not the .XFER instruction, for the .XFER entry point. The .XFER opcode is available for assembly language programmers but must be used with the following constraints.

a. source and desitnation address *must* be direct.

b. an extra location must follow the 105220 (RAM 220B) because the microcode returns to P+2.

c. the location following the call to .XFER is treated as an address and indirects may be traced but nothing is done with this address.

d. the following calling sequence is used:

```
LDA (source address X)
LDB (destination address Y)
RAM 220B
NOP
```

HEWLETT PACKARD

2100 series computers

HP 12907A
fast FORTRAN processor diagnostic

HP Product No. 12907-16003

HP MANUAL PART NO. 12907-90003
MICROFICHE PART NO. 12907-90004

Manual of Diagnostics
January 1974

HP 12907A Fast FORTRAN Processor Diagnostic

The HP 12907A Fast FORTRAN Processor Diagnostic confirms the proper functioning of the HP 12907A Fast FORTRAN Processor (FFP). FFP (as it is installed on the HP 2100 computer) consists of 13 Library Subroutines implemented by microcode on PROM's.

This diagnostic is one of the 2100 series computer system diagnostics executed with the HP 2100 Series Diagnostic Configurator. FFP and this diagnostic are used only on the HP 2100 computer.

Test sequence and test failure reporting to the operator is provided through a teleprinter (if available), through the computer Memory Data Register (MDR), and through the A-register. Operator input is required via the switch register for test options and via the A-register for individual test selection.

LIMITATIONS

All microcode failure types are detected by the diagnostic, except:

1. If the microcode does not return control to the diagnostic program, test validity cannot be assured. This situation results in the cessation of messages to the operator. Pressing HALT on the computer will usually *not* halt the computer. The only remedy is to turn the power off and reload the diagnostic program.

2. If the microcode returns control to the diagnostic program but not to the proper location, the results are meaningless. Reload the diagnostic program to continue the test.

3. If any of the following errors are detected, reload the program.

    a. Any error in Test 0, .GOTO Test.

    b. Errors in Test 1 indicated by the messages E050, E051, E052, or E054.

    c. Errors in Test 2 indicated by the message E060.

    d. Errors in Test 3 indicated by the message E102 or E103.

    e. Errors in Test 11 indicated by message E210.

    f. Errors in Test 12 indicated by message E220.

GENERAL ENVIRONMENT

The general hardware/software environments and system configuration procedures are described in the *HP 2100 Series Diagnostic Configurator (02100-90157).*

Hardware Requirements

1. The diagnostic is run on either a 2100A or 2100S computer, with a minimum of 4K of memory.

2. A paper tape reader is required to load the program only; the teleprinter paper tape reader can be used.

3. A system console teleprinter is optional hardware, but recommended, to communicate test results to operator.

4. The HP 12901 Floating Point Processor must be installed on the computer.

5. The HP 12907 Fast FORTRAN Processor must be installed.

6. The HP 12539 Time Base Generator is optional. If the Time Base Generator is not present, the interrupt checks, which are made for .XADD, .XSUB, .XMPY, and .XDIV subroutines, will be omitted.

Software Requirements

The required software consists of the following binary object tapes:

1. HP 2100 Series Diagnostic Configurator (24296).

2. FFP diagnostic (12907-16003).

Loading is performed using a Binary Loader (memory resident). See the *2100 Front Panel Procedures (5951-1371)* for use of the Binary Loader. The loaders are described in the HP manual *Basic Binary Loader-Basic Binary Disc Loader-Basic Moving Head Disc Loader (5951-1376).*

Operating Procedures

The Operating Procedures section is divided into four parts: Summary, Preparation for Diagnostic Run, Running the Diagnostic, and Diagnostic Messages and Halts.

SUMMARY

1. Insure that the FFP is installed properly.

2. Install a Time Base Generator if the interrupt tests are to be executed.

3. Load the Diagnostic Configurator (24296) using BBL, BBDL, or BMDL.

4. Configure using Diagnostic Configurator. Refer to *HP 2100 Series Diagnostic Configurator* manual for configuration procedure.

5. Load the HP 12907A Fast FORTRAN Processor Diagnostic using the BBL, BBDL, or BMDL.

6. If a Time Base Generator is present, go to step 7. Otherwise, set P-register to 2000_8, press EXTERNAL PRESET and INTERNAL PRESET, then go to step 9.

7. Load the Select Code of the Time Base Generator into bits 0-5 of the switch register, set the P-register to 100_8.

8. Press EXTERNAL PRESET and INTERNAL PRESET, press RUN;

   Result: HALT

   - MDR = 102073_8 if invalid SC; to correct, enter correct Time Base Generator Select Code (10_8 or greater), press RUN.
   - MDR = 102074_8 if valid SC.

9. Set the switch register = the program options (selected from Table 1).

10. Press RUN.

Result: The message

START 2100A-S FFP DIAGNOSTIC

is printed. Automatic execution of all selected tests begins.

Each test is preceded with a test title message. Appropriate error messages are printed for software detected FFP test failures.

Note: Anytime during diagnostic execution that switch 9 is set, test run aborts at the end of the current with HALT (MDR = 102075_8). This gives the user the opportunity to specify a different group of tests.

At the completion of all selected tests, the message

PASS *nnnnnn*

is printed, where *nnnnnn* is the octal pass count. A HALT follows with MDR = 102077_8 and the pass count in the A-register.

11. See section "Restarting Diagnostic," for restarting procedures.

PREPARATION FOR DIAGNOSTIC RUN

Before the tests can be initiated, the user performs the following actions:

- Verify proper installation
- Load the Diagnostic Configurator
- Configure to available system hardware
- Load the diagnostic
- Dump the configuration for later use

Verify Proper Installation

Insure that FFP is installed according to procedures given in the *Installation Manual for the 12907A Fast FORTRAN Processor Hardware Accessory Kit (12907-90001).* If interrupt tests are to be executed, install the Time Base Generator.

Load the Diagnostic Configurator

Using the Binary Loader, load the HP 2100 Series Diagnostic Configurator. Perform the configuration procedure (see "Configure" below) before loading the diagnostic.

Configure

Procedures for inputting the system hardware configuration parameters are found in *HP 2100 Series Diagnostic Configurator (02100-90157)* under "Configuring."

The configuration procedure accepts six groups of parameters. This diagnostic requires only three groups to be defined. They are

- Computer type and options
- Teleprinter as optional system Output B device
- Memory size and type

Enter zero (or the Teleprinter Select Code) for the other three parameters. Configure *Computer Type and Options* for the HP 2100 computer with Floating Point Hardware and the other options installed on the system.

A teleprinter is recommended for configuration as *System Output B (Slow Output) Device.* If no teleprinter is available, configuration input is zero. The user must then rely on HALT codes and the switch register to monitor test sequence and determine the degree of test success.

Load the Diagnostic

Load the HP 12907A Fast FORTRAN Processor Diagnostic using the Binary Loader. The user may verify that the proper diagnostic is loaded by checking memory location 126_8 for the Diagnostic Serial Number = 101010_8.

Dump the Configuration

Using procedures described in the *Diagnostic Configurator* manual, the user may dump memory onto paper tape so that the above configuration procedures need not be repeated. The dumped paper tape provides a configured diagnostic.

RUNNING THE DIAGNOSTIC

Parameter Entry and Program Options

Table 1 holds a summary of switch register options.

If *switch register bit 9* is set, either prior to starting test execution or while running, the diagnostic halts with MDR = 102075_8. The user then selects test *i* to be executed by setting bit *i* of the A-register. Refer to the first page of Test Sections for test section numbers and names.

If *switch register bit 12* is set, the diagnostic will not HALT at the completion of all selected tests and will repeat the diagnostic as configured from the beginning.

Table 1. Switch Register Options

Bits	Function If Set
0 to 5	Select code of the Time Base Generator, if used.
6 to 8	Reserved
9	Abort current tests; halt with MDR = 102075_8; user sets bits of A-register with test selection where bit *i* selects test *i*; e.g. bit 0 set selects test 0, bit 1 set selects test 1, etc.
10	Suppress printing of H-type messages (see Table 3).
11	Suppress printing of E-type messages (see Table 3).
12	Repeat all selected tests.
13	Repeat currently executing test (loop).
14	Suppress error halts (see Table 2).
15	Halt at the end of each test; MDR = 102076_8; A-register contains octal test number of test just executed.

Restarting Diagnostic

- If the operator wishes to repeat the diagnostic as configured, press RUN.
- If the operator wishes to change only the set of tests executed, he does the following:
  1. Set bit 9 of switch register.
  2. Press RUN.
  3. Set A-register bits for tests desired (A-register clear means all tests are executed).
  4. Clear bit 9 of switch register.
  5. Press RUN.

- If the operator wishes to restart diagnostic any time without any parameter change, he does the following:

  1. Set P-register = 2000_8.
  2. Press PRESET EXTERNAL and PRESET INTERNAL.
  3. Press RUN.

- If the operator wishes to restart diagnostic with new Time Base Generator Select Code, he follows the procedure in the section "Summary."

DIAGNOSTIC MESSAGES AND HALTS

The diagnostic communicates to the operator through the teleprinter, halts, or both, based on configuration and switch register settings. Thus messages consist of halt codes (MDR and A-register values) and/or teleprinter text.

Halt Summary

Table 2 lists octal halt codes and their significance. Error halts are suppressed if switch register bit 14 is set.

The "end of test" halt is executed, if switch register bit 15 is set; that is the diagnostic executes a halt instead of proceeding to the next test.

Halt at "end of pass," that is halt at end of execution of all selected sections of the diagnostic, is *suppressed*, if switch register bit 12 is set.

Halt to allow user to make a new test group selection is executed, if switch register bit 9 is set.

Table 2. HALT Codes and Significance

Octal MDR Code	Significance
102030-102060 106000-106062 103000-103021	Error (E) messages 30_8 to 221_8 described in Table 3.
102073	HALT indicating select code input error during "Starting Up" procedure. Input valid select code; press RUN.
102074	Valid select code entry was made; make program option switch register setting; press RUN.
102075	HALT to allow test selection in A-register; make test selection; press RUN.
102076	End of test section HALT; A-register holds test number just completed.
102077	Diagnostic completed; A-register holds octal number of passes completed.
106077	Trap cell HALTs stored in CPU memory locations 2_8 to 77_8; indicates hardware malfunction.

Message Summary

There are two general categories of messages output to the operator: program/operator communication messages and test failure (error) messages. Table 3 lists diagnostic messages ordered by message number. Those that are coded with the letter "H" identify them as communication messages. All error messages are coded with the letter "E." Communication messages are printed, if switch register bit 10 is clear. Error messages are printed if switch register bit 11 is clear.

The test that outputs each message is indicated in the table. The tests are described in this manual under the heading "Test Sections."

"TC" refers to the Test Control program. Otherwise, the numbers refer to the test number.

Table 3. Messages to Operator

Message	Test	Meaning
START 2100A-S FFP DIAGNOSTIC	TC	Diagnostic title message.
TEST *nn*	TC	Indicates to which test a list of error messages, which follow, belongs.
PASS *nnnnnn*	TC	All selected tests of the diagnostic have been completed; *nnnnnn* is the octal number of passes completed; A-register holds the octal number of passes completed, if HALT is invoked.
H030 .GOTO TEST	0	Header message for test 0.
E030 FAILED FOR INDIRECT ADDRESSING	0	The 16 levels of indirect addressing were not properly processed.
E031 FAILED FOR J=0	0	Control did not return to the location pointed to by the first address pointer.
E032 FAILED FOR J=NEG	0	Control did not return to the location pointed to by the first address pointer.
E033 FAILED FOR J>16	0	Control did not return to the location pointed to by the last (16th) address pointer.
E034 FAILED FOR J=8	0	Control did not return to the location pointed to by the 8th address pointer.
H050 .ENTR TEST	1	Header message for test 1.
E050 FAILED FOR ACTUAL NR OF PARAM. < ALLOWED NR	1	Subtest 1 failed.
E051 FAILED FOR ACTUAL NR OF PARAM. =ALLOWED NR	1	Subtest 2 failed.
E052 FAILED FOR ACTUAL NR OF PARAM. > ALLOWED NR	1	Subtest 3 failed.
E053 NO CHECK ON MEM PROT VIOLATION	1	Microcode does not test for Memory Protect violation.
E054 RETURN ADDRESS NOT STORED IN CORRECT LOCATION	1	Microcode failed to store return address in correct location.
E055 RETURN ADDRESS NOT IN A-REG.	1	A-register does not contain the return address of the subroutine, which called the routine .ENTR.
E056 INCORRECT ADDR. IN B-REG.	1	B-register should contain the address of the first location into which *no* parameter address was stored. In the case that the number of actual parameters is equal to or larger than the number of allowed parameters, the B-register should contain the address of the last allowed parameter location +1.

Table 3. Messages to Operator (Continued)

Message	Test	Meaning
H060 .ENTP TEST	2	Header message for test 2.
E060 FAILED FOR ACTUAL NR OF PARAM. < ALLOWED NR	2	Test failed.
H100 .SETP TEST	3	Header message for test 3.
E100 A-REG. NOT=0 UPON RETURN	3	A-register does not contain the initial value (zero) upon return from microcode.
E101 B-REG. DOES NOT CONTAIN LAST ADDRESS+1 UPON RETURN	3	B-register does not contain the correct address upon return from microcode.
E102 INCORRECT VALUE STORED	3	Microcode stored an incorrect value or did not store anything in a certain location.
E103 MORE LOCATIONS FILLED THAN REQUESTED	3	Correct values were stored in the requested locations, but the next locations were also altered.
E104 NO CHECK ON MEM PROT VIOLATION	3	Microcode does not check on Memory Protect violation.
H110 . .MAP TEST	4	Header for test 4.
E110 DATA ERROR ACT *xxxxxx* EXP *yyyyyy*	4	Invalid data=*xxxxxx* was returned by microcode; data should have been *yyyyyy*; when the error Halt occurs (MDR=106010), the A-register and B-register contain the actual and expected data, respectively.
H120 SNGL TEST	5	Header for test 5.
E120 DATA ERROR ACT *xxxxxx xxxxxx* EXP *yyyyyy yyyyyy*	5	Microcode returned the single precision number *xxxxxx xxxxxx* instead of *yyyyyy yyyyyy*; when the error Halt occurs (MDR=106020), the A-register holds the first word and the B-register holds the second word of the returned data; pressing RUN will cause a Halt (MDR=107000) where A-register holds the first word and B-register holds the second word of the expected data.
E121 OVERFLOW NOT SET	5	The number processed by the microcode caused an underflow or overflow condition, but the Overflow register was not set.
E122 FENCE ADDR NOT SAVED	5	The Memory Protect Fence address was not saved and restored.
E123 OVERFLOW SET	5	The Overflow should not be set.
H130 DBLE TEST	6	Header for test 6.

Table 3. Messages to Operator (Continued)

Message	Test	Meaning
E130 DATA ERROR ACT *xxxxxx xxxxxx xxxxxx* EXP *yyyyyy yyyyyy yyyyyy*	6	The extended precision number *xxxxxx xxxxxx xxxxxx* was returned instead of *yyyyyy yyyyyy yyyyyy*; when the error Halt occurs (MDR=106030), the A- and B-registers contain the first and second words of returned data; pressing RUN will cause another Halt (MDR=107000) to occur where the A-register holds the third word of the actual data; pressing RUN again will cause another Halt (MDR= 107001) where the A- and B-registers hold the first and second words of the expected data; pressing RUN yet again will cause the final Halt in the series (MDR= 107002) where the A-register holds the third word of the expected data.
E131 NO CHECK ON MEM PROT VIOLATION	6	Microcode does not check for memory protect violation.
H140 .XADD TEST	7	Header for test 7.
E140 DATA ERROR ACT *xxxxxx xxxxxx xxxxxx* EXP *yyyyyy yyyyyy yyyyyy*	7	The extended precision number *xxxxxx xxxxxx xxxxxx* was returned instead of *yyyyyy yyyyyy yyyyyy*; when the error Halt occurs (MDR=106040), the A- and B-registers hold the first and second word of actual data; pressing RUN will cause another Halt (MDR=107000) to occur where the A-register holds the third word of the actual data; pressing RUN again will cause another Halt (MDR= 107001) where the A- and B-registers hold the first and second words of the expected data; pressing RUN yet again will cause the final Halt in the series (MDR= 107002) where the A-register holds the third word of the expected data.
E141 OVERFLOW NOT SET	7	The number processed by microcode caused an underflow or overflow condition, but the Overflow register was not set.
E142 NOT INTERRUPTIBLE	7	Microcode does not check for interrupt.
E143 OVERFLOW SET	7	The overflow should not be set.
H150 .XSUB TEST	8	Header for test 8.
E150 DATA ERROR ACT *xxxxxx xxxxxx xxxxxx* EXP *yyyyyy yyyyyy yyyyyy*	8	The extended precision number *xxxxxx xxxxxx xxxxxx* was returned instead of *yyyyyy yyyyyy yyyyyy*; when the error Halt occurs (MDR=106050), the A- and B-registers hold the first and second words of the actual data; pressing RUN will cause the same sequence of Halts described in error messages E130 and E140, showing the rest of the actual and expected values.
H160 .XMPY TEST	9	Header message for test 9.

Table 3. Messages to Operator (Continued)

Messages	Test	Meaning
E160 DATA ERROR ACT *xxxxxx xxxxxx xxxxxx* EXP *yyyyyy yyyyyy yyyyyy*	9	The extended precision number *xxxxxx xxxxxx xxxxxx* was returned instead of *yyyyyy yyyyyy yyyyyy*; when the error Halt occurs (MDR=106060), the A- and B-registers hold the first and second words of the actual data; pressing RUN will cause the same sequence of Halts described in error messages E130 and E140, showing the rest of the actual and expected values.
E161 OVERFLOW NOT SET	9	The operation caused an overflow or underflow condition, but the Overflow register was not set.
E162 NOT INTERRUPTIBLE	9	Microcode does not check for interrupt.
H200 .XDIV TEST	10	Header message for test 10.
E200 DATA ERROR ACT *xxxxxx xxxxxx xxxxxx* EXP *yyyyyy yyyyyy yyyyyy*	10	The extended precision number *xxxxxx xxxxxx xxxxxx* was returned instead of *yyyyyy yyyyyy yyyyyy*; when the error Halt occurs (MDR=103000), the A- and B-registers hold the first and second words of the actual data; pressing RUN will cause the same sequence of Halts described in messages E130 and E140.
E201 OVERFLOW NOT SET	10	The operation caused an overflow or underflow condition, but the Overflow register was not set.
H210 .DFER TEST	11	Header message for test 11.
E210 FAILED	11	Microcode failed to execute the operation correctly.
E211 NO CHECK ON MEM PROT VIOLATION	11	Microcode does not check for Memory Protect violation.
H220 .XFER TEST	12	Header message for test 12.
E220 FAILED	12	Microcode failed to execute the operation correctly.
E221 RETURN AT INCORR LOC	12	Microcode returned to wrong memory location.

Test Sections

The 12907A Fast FORTRAN Processor Diagnostic provides thirteen tests to exercise each of the thirteen microcoded subroutines, which constitute the Fast FORTRAN Processor. The general method is to call each subroutine to exercise the functions.

Each test section listed below tests one microcoded subroutine.

Test Number	Title
0	.GOTO
1	.ENTR
2	.ENTP
3	.SETP
4	..MAP
5	SNGL
6	DBLE
7	.XADD
8	.XSUB
9	.XMPY
10	.XDIV
11	.DFER
12	.XFER

Note: Refer to the Relocatable Subroutines (02116-91780) for each of the following subroutines.

.GOTO — TEST 0

Microcode for the .GOTO subroutine is tested for five different parameter groups and for 16 levels of indirect addressing.

.ENTR — TEST 1

Microcode for the .ENTR subroutine is tested for the actual number of parameters less than, equal to, and greater than the allowed number of parameters. Verifies that microcode is testing for Memory Protect Violation when storing the subroutine return address and when transferring the parameter addresses.

.ENTP — TEST 2

Since this subroutine is simply another entry point into the .ENTR subroutine, one test is performed to verify that the microcode executes properly for the actual number of parameters less than the allowed number of parameters.

.SETP — TEST 3

Microcode for the .SETP subroutine is tested for one case. It is also verified that the microcode tests for Memory Protect Violation when storing the values.

..MAP — TEST 4

Microcode for the ..MAP subroutine is tested for several elements of a two-dimensional array and then for several elements of a three-dimensional array.

SNGL — TEST 5

Microcode for the SNGL subroutine is tested for several extended precision numbers, including those which cause underflow and overflow. It is verified that the microcode saves and restores the Memory Protect Fence register address.

DBLE — TEST 6

Microcode for the DBLE subroutine is tested for several single precision numbers. It is verified that microcode tests for Memory Protect Violation when storing results.

.XADD — TEST 7

Microcode for .XADD subroutine is tested for several operands, including those which cause underflow and overflow conditions. The interruptibility of the microcode is tested.

.XSUB — TEST 8

Microcode for .XSUB subroutine is tested for several operands.

.XMPY – TEST 9

Microcode for .XMPY subroutine is tested for several operands, including those which cause underflow and overflow conditions. The interruptibility of the microcode is tested.

.XDIV – TEST 10

Microcode for .XDIV subroutine is tested for several operands, including those which cause an underflow and overflow conditions. Division by zero is tested.

.DFER – TEST 11

Microcode for .DFER subroutine is tested for one case. It is verified that the microcode tests for Memory Protect Violation during transfer.

.XFER – TEST 12

Microcode for .XFER subroutine is tested for one case.